On a Collision Course: Unveiling Wireless Attacks to the Aircraft Traffic Collision Avoidance System (TCAS)

Giacomo Longo, *DIBRIS, University of Genova;* Martin Strohmeier, *Cyber-Defence Campus, armasuisse S + T;* Enrico Russo, *DIBRIS, University of Genova;* Alessio Merlo, *CASD, School of Advanced Defense Studies;* Vincent Lenders, *Cyber-Defence Campus, armasuisse S + T*

https://www.usenix.org/conference/usenixsecurity24/presentation/longo

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

On a Collision Course: Unveiling Wireless Attacks to the Aircraft Traffic Collision Avoidance System (TCAS)

Giacomo Longo
DIBRIS
University of Genova, Italy
giacomo.longo@dibris.unige.it

Martin Strohmeier
Cyber-Defence Campus
armasuisse S + T, Switzerland
martin.strohmeier@armasuisse.ch

Enrico Russo
DIBRIS
University of Genova, Italy
enrico.russo@unige.it

Alessio Merlo
CASD
School of Advanced Defense Studies, Italy
alessio.merlo@ssuos.difesa.it

Vincent Lenders
Cyber-Defence Campus
armasuisse S + T, Switzerland
vincent.lenders@armasuisse.ch

Abstract

Collision avoidance systems have been a safety net of last resort in aviation since their introduction in the 1980s. Through constantly refined safety procedures and hard lessons learned from mid-air collisions, the Traffic Collision Avoidance System (TCAS) II Version 7.1 has become the global standard, significantly improving safety in a fast-growing field.

Despite this safety record, TCAS was not designed with security in mind, even in its newest versions. With the rise of software-defined radios, security researchers have shown many wireless technologies in aviation and critical infrastructures to be insecure against radio frequency (RF) attacks. However, while similar attacks have been postulated for TCAS with its built-in distance measurement, all attempts to execute them have failed so far.

In this paper, we introduce the first working RF attacks on TCAS. We demonstrate how to take full control over the collision avoidance displays and create so-called Resolution Advisories (RAs) of arbitrary aircraft on a collision course. We build the necessary tooling using commercial off-the-shelf hardware, creating sufficient conditions for the attacker to spoof colliding aircraft from a distance of up to 4.2 km.

We evaluate this and further attacks extensively on a live, real-world, certified aircraft test system and discuss potential countermeasures and mitigations that should be considered by aircraft and system manufacturers in the future.

1 Introduction

In aviation, collision avoidance is a critical safety measure, with the TCAS standing out as the last line of defense against mid-air disasters. As advancements in aviation technology have contributed to an exponential surge in air traffic globally, there is a critical need for robust, reliable, and secure systems capable of serving the ever-growing number of passengers and flights. TCAS, operational since the 1980s, has demonstrated its effectiveness through continuous safety enhancements and now has an impressive safety record, preventing many mid-air collisions every year [44, 50].

On the most basic level, TCAS operates by detecting and tracking nearby aircraft, displaying them in the cockpit and providing pilots with visual and aural advisories and resolutions to avoid potential conflicts in time. Crucially, the system is independent of Air Traffic Control (ATC) and can thus catch human errors made by pilots or controllers as a last resort. Consequently, when TCAS malfunctions or is accidentally deactivated, the repercussions can be significant. Such incidents have been ranging from very close calls [33] to tragic mid-air collisions [8]. Pilots erroneously disregarding the advisories of TCAS has previously led to fatal incidents, too [14].

Beyond such traditional safety issues, glaring novel security vulnerabilities have been exposed in many avionics and communication systems in the academic literature of the last decade. The evidence has shown most of the employed wireless technologies in aviation to be exploitable in controlled laboratory environments using accessible Software-Defined Radios (SDRs) and software tools [10,41–43,45,46]. Notwithstanding their potentially major impact, these vulnerabilities seem straightforward in hindsight: as there are few to no cryptographic security measures, and encoding and modulation are known, full control over the channel is dependent only on an attacker's sending power.

However, while TCAS is using some of the same underlying vulnerable wireless channels, it has so far resisted any proof-of-concept attack even in the laboratory. This is due two main reasons: first, TCAS has a strong physical-layer component, whereby dynamic, fast-moving, targets use interrogations and responses for speed-of-light distance measurements. Second, the complexity of TCAS-specific communication and its associated state machines, along with the need for more advanced SDR setups than those used for analyzing other protocols, results in a lack of existing tooling for this purpose.

Indeed, many researchers have postulated the possible existence of vulnerabilities in TCAS [5, 17, 18, 48] and even examined their potential impact on aircraft and pilots in flight simulators [47]. However, unlike attacks on systems such as ADS-B, actually executing an attack on TCAS has proven hard, with all previous attempts coming up short.

The present research thus marks a significant breakthrough, employing commercial off-the-shelf hardware to delve into the theoretical and practical aspects of a real-world TCAS attack. We explore the challenges and complexities that have made successful exploitation impossible so far and construct the necessary tools, models and laboratory setups for comprehensive, yet safe and ethical, attack execution and evaluation.

Given the significance of TCAS and its presence on most commercial jetliners [1, 23], the goal of this research is to understand the attack requirements against this crucial system. In combination with existing literature on the impact of such attacks on collision avoidance systems, we hope that our work will lead to increased awareness within the aviation community and the development of adequate technical or procedural countermeasures and mitigations.

In summary, our contributions are as follows:

- We present the first working RF attacks on TCAS. Most notably, we take arbitrary control over the collision avoidance displays and create resolution advisories of aircraft on collision course at will.
- We build novel tooling, modeling, and a certified testbed needed in order to truly understand and conduct live attacks on physical-layer collision avoidance systems.
- We experimentally, yet safely, evaluate the feasibility and working parameters of three different attacks on TCAS with our testbed setup.

2 Background

2.1 Mode S Communication

Mode S, short for "Mode Select", is a communication protocol used in air traffic control and aviation. It enables aircraft to exchange information with ground-based radar systems and other aircraft through on-board radio transponders. Its *select* feature sets it apart from previous communication methods. This feature is the inclusion of a globally unique 24-bit address in each message, assigned by the International Civil Aviation Organization (ICAO) according to a regional delegation scheme [20]. Such an address identifies the sender or recipient of each message.

Physical layer. Mode S specifies two distinct packet-based communication channels over radio links. The first, the *uplink*, uses Differential Binary Phase Shift Keying (D-BPSK) over the 1030 MHz frequency, running at a data rate of 4 Mbit/s. The second, namely *downlink*, uses pulse position modulation over the 1090 MHz frequency, with a data rate of 1 MBit/s. Each packet consists of a fixed amplitude modulated preamble followed by either 56 or 112 bits of modulated payload [39].

Data format. Each message is distinguished based on the channel used and the initial five bits, known as the *format*. Uplink channel messages are denoted as UF=x, with "x" representing the format. Similarly, downlink channel messages are designated as DF=x. For any given DF=x or UF=x, the interpretation of its subsequent bits remains consistent, with the standard [39, 2.2.14.1] determining the position, length, and names for multiple fields. However, it is essential to note that while a particular field may appear in multiple messages, it does not necessarily mean that the same type of information will consistently be present within it. As an example, the four-bit *Reply Information* (RI) field in *Short air-air surveillance* (DF=0) responses can either signify the maximum velocity range of the queried aircraft (if $RI < 8$) or describe the type of installed TCAS (if $RI \geq 8$).

Additionally, every packet ends with a 24-bit CRC-based parity. Depending on the message's channel and format, this parity can either be XORed with 24 ones or undergo a channel-specific mathematical transformation, encoding a 24-bit ICAO address (uniquely identifiying the aircraft transponder) in conjunction with the parity. This method conserves bits that would otherwise be used for adding the address to each message. However, it requires recipients of the mathematically adjusted parity field to possess prior knowledge of the address, acquired from earlier messages that included the parity XORed with 24 ones.

Protocols. Mode S integrates two primitives that its upper-layer protocols use: *broadcasts* and *interrogations*. Broadcasts do not require a response. Interrogations are sent over uplink and are associated with a response on downlink 128μs after reception. In the case of interrogations, a message with format specifier UF=x is often associated with its DF=x counterpart.

These primitives are leveraged to implement multiple functionalities such as *Air Traffic Control Radar Beacon System/Mode Select (ATCRBS/Mode S)* [11], *Automatic Dependent Surveillance - Broadcast* (ADS-B) [21], and Traffic Collision Avoidance System (TCAS) [37], the subject of this study.

2.2 The Traffic Collision Avoidance System

TCAS is a standard aircraft collision avoidance system that plays a crucial role in modern aviation safety. Originally introduced as TCAS I by the Radio Technical Commission for Aeronautics (RTCA), this system was created to offer pilots proximity alerts regarding nearby aircraft using radio communication [36]. Subsequently, an upgraded version known as TCAS II was introduced, adding the capability of suggesting evasive maneuvers to actively avoid potential mid-air collisions [37]. Both iterations of the standard are internationally ratified by ICAO under the names Airborne Collision Avoidance System (ACAS) I and ACAS II [22]. These ACAS standards are equivalent to their TCAS counterparts

in terms of functionality and purpose [19]. TCAS is a widely adopted standard, not just due to its effectiveness in enhancing flight safety but also because of legal mandates requiring its installation in almost every commercial aircraft [1, 13, 23]. Interoperable TCAS II equipment is sold by four US-based vendors. TCAS is set to be superseded by ACAS X, which is currently still in the standardization phase and not yet operational. Even after the eventual introduction of ACAS X in the 2030s, downwards compatibility with TCAS will ensure that the latter will remain in use for decades to come.

Surveillance. A primary feature of TCAS is *surveillance*. This function allows pilots to see on their instruments the relative positions and altitudes of other aircraft equipped with Mode S transponders. To find traffic in the vicinity, TCAS listens for spontaneous broadcasts, namely *squitters*, keeping track of the ICAO addresses contained in them. TCAS periodically interrogates the found aircraft to track their altitude, bearing, and range. In particular, it extracts altitude from the message content replied by the interrogated aircraft. Bearing and range are instead measured independently using signal processing techniques. For bearing, the TCAS equipment must incorporate a radio direction finding device capable of estimating the angle of arrival of the interrogation response. In contrast, the range calculation relies on the fixed delay of $128\mu s$ between the reception and the initiation of each Mode S interrogation response.

$$R = \frac{c}{2}(\Delta T - 128 \cdot 10^{-6}) \tag{1}$$

Equation 1 details the method for determining range. It involves multiplying the speed of light, denoted as c, with subtraction between round trip time ΔT and the expected delay. This result is then halved to account for the round-trip journey of the radio wave to the interrogated aircraft and back.

Traffic Advisory (TA). The collision avoidance capability of TCAS extends beyond simply displaying nearby aircraft. In fact, it continuously monitors each tracked aircraft to alert the crew of potentially dangerous encounters preemptively. Whenever it detects an intruding aircraft as a threat, TCAS issues a Traffic Advisory (TA), an urgent alarm highlighting an elevated risk of in-flight collision. Triggering a TA requires a potential threat to satisfy specific conditions on altitude and range:

$$ADOT = \dot{A} \cdot \mathrm{sgn}(\Delta A)$$
$$(\Delta A < THR_{TA}) \bigvee \tag{2a}$$
$$\left(ADOT \geq -1\text{ft/s} \bigwedge -\frac{\Delta A}{ADOT} < TAU_{TA}\right) \tag{2b}$$

Equation 2 reports the condition on altitude in Disjunctive Normal Form (DNF). There, ΔA is the difference in altitude between the TCAS-equipped aircraft and the intruder, with $\dot{A}$ being its derivative w.r.t. time. The first term (2a) merely checks whenever the altitude difference is less than THR_{TA}. Instead, the second term (2b) applies whenever the altitudes of the aircraft are converging, comparing the time to the closest vertical point of approach with a threshold TAU_{TA}.

$$\left(R\dot{R} \leq H1_{TA} \bigwedge \dot{R} > 10\text{ft/s} \bigwedge R \leq DMOD_{TA}\right) \bigvee \tag{3a}$$

$$\left(\dot{R} \leq 10\text{ft/s} \bigwedge \frac{\frac{DMOD_{TA}^2}{R} - R}{\min(-10\text{ft/s}, \dot{R})} \leq TAU_{TA} \bigwedge R > DMOD_{TA}\right) \bigvee \tag{3b}$$

$$\left(\dot{R} \leq 10\text{ft/s} \bigwedge R \leq DMOD_{TA}\right) \tag{3c}$$

Equation 3 reports the condition on range in DNF. It contains the currently measured range R and its time derivative, i.e., closure rate, $\dot{R}$. The first term (in 3a) applies whenever the aircraft are approaching each other, the tracked range is less than $DMOD_{TA}$, and the product between range and closure rate is less than $H1_{TA}$. The second (3b) and third (3c) terms apply to the case of diverging or non-rapidly approaching intruders. Similar to the previous case, those last two terms trigger a TA either when the horizontal time to the closest point of approach is less than TAU_{TA} or when a range measurement is closer than the $DMOD_{TA}$ threshold value.

Table 1 details the coefficients and thresholds associated with triggering a TA, as they appear in the standard. It is worth noting that the values vary according to the current Sensitivity Level (SL). We discuss SL in the following.

Resolution Advisory (RA). RAs are one of the main innovations introduced with the second version of TCAS. They extend TAs with facilities for automatically resolving conflicts by issuing commands, called RAs, to the cockpit. These commands direct the pilots to either adopt a specific vertical attitude (for example by climbing or descending) or to modify the ongoing collision avoidance action, such as leveling off or adjusting the ascent/descent rate. TCAS will continue to issue and update those commands until resolution of the conflict.

The decision of issuing a RA is taken by TCAS once an ongoing TA risk increases to an even more critical level. As such, RAs are triggered by repeating the checks found in Equations 2 and 3 but replacing each coefficient value with the one from its non-TA counterpart. For example, $H1_{TA}$ in Equation 3a gets replaced by $H1$. In addition, RAs can only be issued whenever the intruder vertical separation at the closest point of approach V_{CPA} is less than the value $ALIM$.

$$V_{CPA} \approx \begin{cases} \Delta A + (\dot{A} - \dot{A_O})\frac{R}{\dot{R}} & \text{if } \dot{R} \gg 0 \\ \Delta A & otherwise \end{cases} \tag{4}$$

For brevity, we approximate the procedure for calculating V_{CPA} [38, 3.96] with Equation 4 where $\dot{A_O}$ is the intruder altitude time derivative.

The choice of which action to issue as RA can be performed either autonomously or collaboratively, depending on the intruder equipment. TCAS II systems announce their presence by periodically emitting an additional dedicated TCAS squitter. An intruder is considered capable of coordination after reception of such squitter and when DF=0 responses to the surveillance function indicate an "On-board TCAS with vertical-only resolution capability" in their RI field.

If TCAS is unsure about the availability of coordination, it unilaterally chooses RA contents according to the intruder-tracked position and attitude. For instance, TCAS issues *climb* RAs against lower altitude targets.

Instead, intruders capable of coordinating encounters negotiate by choosing their action as per the uncoordinated case and exchanging *Long Special Surveillance interrogation* (UF=16) messages instructing the other aircraft to not pick a conflicting resolution, e.g., a descending aircraft will instruct the other to not do so. In a disagreement, the TCAS associated with the aircraft with the highest ICAO address will pick an action opposite to the one chosen by the other system.

Sensitivity Level (SL). TCAS relies on standardized thresholds to classify the status of intruders. The values of these thresholds depend on the currently set Sensitivity Level (SL), a value ranging from 1 to 7. SL=1 is a special level corresponding to a disablement of all TCAS functions, including surveillance. SL=2 corresponds to the *TA Only* mode of operation in which TCAS does not issue any RAs. SL values of 3 to 7 are instead associated with the full functionality, with higher sensitivities corresponding to larger areas (so-called *protected volumes*) being considered dangerous by the system.

On most TCAS units, pilots can only choose between three settings: STBY (standby), TA, and TA/RA. STBY and TA correspond to fixed SLs of 1 and 2, respectively. TA/RA mode automatically selects a SL from 2 to 7 depending on the current altitude and the standardized values in Table 1. During flight, standard procedures suggest that pilots should always set the TCAS mode selector to "TA/RA" [12].

3 Related work

Past literature has discussed the possibility of vulnerabilities in TCAS, as outlined in several surveys on aviation cybersecurity [16,30,31,49]. Overall, the findings agree that there are potential issues originating from the unauthenticated nature of TCAS compounded by the absence of encryption.

Hannah et al. [18] described the threat landscape within TCAS, including the capabilities of malicious actors and the potential range of attacks. Among the threat taxonomy, they classify the ability of adversaries to inject aircraft into a target's traffic display as *False Injection* and the capacity of denying, degrading, or destroying communications as *Denial-of-Service (DoS)*. Importantly, the authors emphasize the complexity of executing such attacks by linking them to individuals possessing extensive skill sets, such as Advanced Persistent Threats (APTs) or criminal organizations.

Previous literature has explored proof of concept for false injection attacks, mostly within simulated environments. Here, we discuss these representative studies.

Graziano [15] presents an instance of an attack using SDRs aimed at an operational TCAS device. The implementation highlights the challenge of managing response times. The best latency achieved is around 606*ms*, far exceeding the threshold for an effective attack by over three orders of magnitude. Indeed, the author mentions that this latency corresponds to an *intruder positioned approximately* 908*km away*.

Berges et al. [4, 5] design an attack using SDRs, targeting a model that emulates functionalities of the TCAS system's state machine. This approach utilized SDR technology to create a partially simulated environment, distinct from direct engagement with an actual TCAS. Notably, their model does not consider the response time constraint for range estimation.

Table 1: TCAS thresholds [37,38][1].

Sensitivity Level		2	3	4	5	6	7
Altitude	[ft]·10^3	<1	<2.35	<5	<10	<20	>42
$H1$	[nm^2/s]	n/a	.002	.00278			.004
$H1_{TA}$	[nm^2/s]	.004	.004	.005	.006	.006	.006
DMOD	[nm]	n/a	0.20	0.35	0.55	0.80	1.10
DMOD$_{TA}$	[nm]	0.30	0.33	0.48	0.75	1.00	1.30
TAU	[s]	n/a	15	20	25	30	35
TAU$_{TA}$	[s]	20	25	30	40	45	48
$ZTHR$	[ft]	n/a	600				700
$ZTHR_{TA}$	[ft]	850					
$ALIM$	[ft]	300		350	400	600	700

Table 2: Summary of quantities found in Section 2.2.

Quantity	Description
A	OA altitude
$\dot{A}$	OA altitude derivative
A_o	IA altitude
$\dot{A_o}$	IA altitude derivative
ΔA	Altitude difference $= A - A_o$
$ADOT$	Direction-weighted IA altitude derivative [38, 3.60]
R	IA range
$\dot{R}$	IA range derivative
V_{CPA}	Vertical separation at Closest Point of Approach

OA = Own Aircraft, IA = Intruding Aircraft.

[1] SI units: 1 nautical mile (nm) is 1852 m, 1 foot (ft) is 0.3048 m.

Lomas et al. [28] presented at DEFCON 28 the effects of radio frequency spoofing attacks against TCAS by adding multiple traffic aircraft to an Airbus flight deck simulator scenario. They emphasize that in a realistic environment, TCAS is *difficult to spoof* due to its reliance on time of flight information to determine the distance between aircraft.

Hannah et al. [17] examine attacks in simulated environments to identify what ranges, altitudes, and relative bearings are most vulnerable to false injections.

Smith et al. [47] demonstrate the significant impact of wireless attacks on pilots, including TCAS false injections, by implementing and analyzing malicious flight simulator scenarios. They also simulate collision avoidance attacks against ACAS X [48], and identify a theoretical success rate ranging from 44% to 79%, with an average deviation of 590 feet.

Regarding DoS attacks, past work [16, 17] hypothesizes about potential implementations to leverage electronic warfare tactics, specifically spot jamming techniques.

To the best of our knowledge, no prior research has successfully proven and executed any practical attacks against an airworthiness certified [24] TCAS. Going further, we are also the first to show how to arbitrarily position an intruder ghost aircraft in a controlled manner by effectively managing the complexity of range estimation. Moreover, this is the first work introducing a DoS attack that uniquely exploits features of the TCAS without depending on electronic warfare tactics.

4 Threat Model and Challenges

4.1 Threat Model

In this paper, we consider an attacker whose goal is to increase the probability of a safety incident significantly.

We assume that the aircraft under attack is equipped with a standard-compliant ACAS/TCAS II device operating in TA/RA mode, supporting a flight crew that has been trained to operate it correctly and follow its advisories. In addition, we assume that the attacker has access to a (fixed) location from which they can receive/transmit radio signals targeted at the victim aircraft.

We consider an adversary who is a professional actor capable of gathering solid knowledge for generating or testing a novel attack. Resource-wise, they can access moderately-priced ($\approx 10,000$\$) Commercial Off-The-Shelf (COTS) hardware components such as SDRs, signal amplifiers, antennas, and powerful computers. Based on these capabilities and potential motivations, we identify three main threat actors: terrorists, activists, and nation states. Terrorists primarily aim to instill fear and broadcast their extremist ideologies. Their interest in aviation can be attributed to the potential for causing significant economic disruption, impeding international travel, or even loss of life, and leveraging the visibility of such attacks to convey their extremist ideology to the general public. Activists, sometimes known as "hacktivists" in the digital realm, are driven by a burning desire to spotlight perceived injustices and agitate for societal change. Activists have previously targeted the private aviation sector to demand political action against highly polluting short flights and the usage of private jets. Finally, nation states might be interested in compromising another country's economic stability from a commercial aviation perspective and obtaining a logistic or direct advantage in military theaters as TCAS is standard in governmental and even military aircraft.

4.2 Challenges

TCAS is a public and well-specified standard with theoretically known vulnerabilities. Graziano [15] and Berges et al. [4] illustrate a comprehensive understanding of transmitting and receiving the TCAS protocol via SDRs. However, despite this theoretical knowledge, there is no proof that an attacker, even with the capabilities of a professional actor, can develop a practical, SDR-based, COTS solution capable of interacting with and exploiting an actual TCAS to fulfill malicious intentions as outlined in our threat model.

According to the literature the primary reason for this absence are the complex requirements of an attack implementation that effectively manages the range estimation of TCAS, which serves as a de facto physical-layer security feature.

To meet this timing constraint, attackers face several challenging requirements. Firstly, they must develop a high-performance application capable of receiving signals on the uplink frequency, demodulating, decoding, computing a response, modulating that response, and then transmitting it on the downlink frequency — all within a stringent timeframe of less than *128 microseconds* (Equation 1).

To solve the range spoofing problem (Equation 5) and ensure the practicality of the attack, the actual response time must be even shorter than this challengingly short period. This duration is significantly shorter than the typical response times of programs running on general-purpose operating systems.

Secondly, the precision of these timings is paramount. They must be accurate within hundreds of nanoseconds and maintain coherence across uplink and downlink frequencies. In addition, the continuity of this process is critical: it must operate uninterrupted over a long period for the intended effects of the attack to become apparent.

Achieving such precise timing is possible with specialized and purpose-built hardware solutions but poses significant difficulties for SDRs that rely on general-purpose processors. However, a custom hardware solution created by expert engineers would necessitate resources and time that surpass those available to even relatively resourceful entities. For this reason, an additional challenge is to identify the COTS hardware that can meet the requirements of these attacks and to take full advantage of optimizations in the software layer.

5 Attacks on TCAS

In the following, we present three different attacks on TCAS: traffic advisory injection, resolution advisory injection, and a simple, but effective denial-of-service attack.

5.1 Traffic Advisory Injection

This first attack involves deceiving the TCAS device about the existence of an intruder that is endangering the victim aircraft, prompting it to issue an alarm to alert the crew.

Recalling the functionality of TCAS, malicious actors can accomplish this goal by (i) getting tracked by the victim TCAS, and (ii) ensuring that the injected intruder position satisfies the conditions for a TA.

We detail these two phases below, denoted as *tracking* and *positioning*. Finally, we demonstrate the consequences of an attacker-initiated TA on the traffic display.

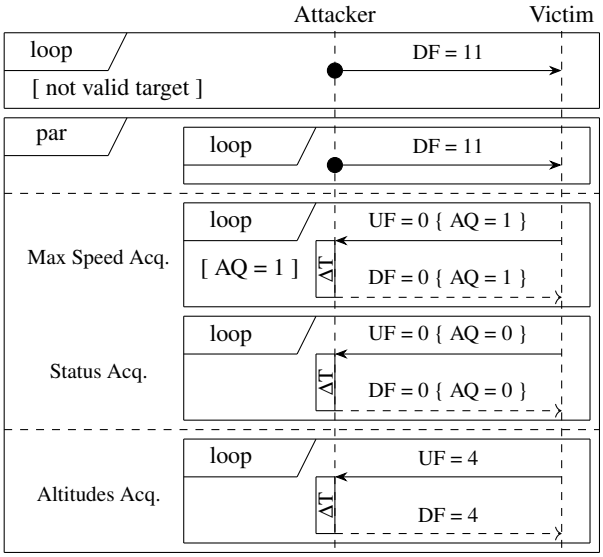

Figure 1: Minimal TCAS surveillance tasks.

Tracking. We consider an attacker implementing a protocol subset roughly corresponding to the air-to-air protocols found in a Level 1 Mode-S transponder [39, 2.2.18]. We also assume the attackers choose an identity for the spoofed aircraft, i.e., an ICAO address. The address is required to initiate interactions with the victim and remains fixed during the attack.

Figure 1 shows the sequence of interactions and messages attackers need to exchange with the victim to enable tracking of a spoofed aircraft.

Operations begin with the attackers emitting *All-call reply* (DF=11) squitters, which broadcast the identity of the spoofed aircraft. This phase loops until the victim adds the spoofed aircraft to its surveillance routines. The standard does not prescribe a specific quantity of squitters but hints at two emissions within its implementation suggestions [37, A.8].

After the victim's TCAS is aware of the spoofed aircraft, operations proceed with three parallel interactions. The first consists of periodically broadcasting DF=11 squitters to keep the surveillance routines active on the victim.

The rest of the interactions are interrogations from the victim to the attacker. In detail, the second interaction comprises two phases. In the first phase, the victim uses *Short air-air surveillance* (UF=0) interrogations to acquire the maximum speed of the spoofed aircraft. This phase concludes when the *Acquisition* (AQ) field of the interrogations is set to 1. In the second phase, throughout the entire attack, the victim monitors the status of the spoofed aircraft. It uses UF=0 interrogations to assess if the spoofed aircraft is airborne.

Finally, the third interaction responds to *Surveillance altitude request* (UF=4) interrogations, continuously updating the injected aircraft altitude on the victim's TCAS.

After completing one round of status and altitude acquisition, the victim initiates tracking of the spoofed aircraft. The victim continues displaying the track on the screen as long as the attacker responds correctly to the victim's interrogations.

With each interrogation, the range of the spoofed aircraft gets updated based on the round-trip time, as previously described (see Section 2.2).

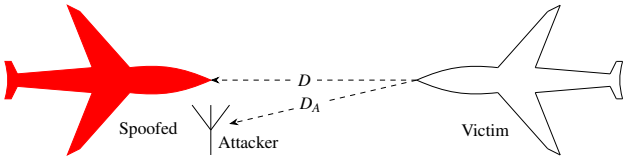

Figure 2: Range spoofing problem.

Positioning. Triggering a TA requires placing the spoofed aircraft such that its measured altitude and range satisfy the conditions described in Section 2.2.

Recalling Equation 2a, responding with any altitude closer than THR_{TA} w.r.t. the victim's altitude suffices in satisfying the altitude condition. The condition on the range instead distinguishes two cases, depending on the spoofed aircraft's closure rate. Placing an aircraft at an almost fixed and less than $DMOD_{TA}$ range suffices in triggering a TA (Equation 3c). Otherwise, the range must evolve over time to satisfy either Equation 3a or Equation 3b.

As higher sensitivity levels are associated with higher likelihoods of generating TAs, attackers can actively ascertain the current SL to gauge which values to use in the formulas. This is performed by sending out a UF=0 interrogation and retrieving the victim's currently configured value in the SL field found in the associated DF=0 response. Otherwise, they can assume the worst-case scenario of SL=2.

For satisfying the altitude condition, attackers modify the *Altitude Code* (AC) field contained within *Surveillance altitude reply* (DF=4) responses. This gives them complete and precise control over its value.

Manipulation of the range is instead performed by altering the interrogations' round trip time ΔT (see Equation 1). Figure 2 depicts the range spoofing problem. There, attackers are positioned at a distance D_A w.r.t. the victim aircraft and aim to inject an aircraft perceived to be at a *different* range D. Under the assumptions from Section 4.1, attackers cannot alter their position throughout the attack execution.

Recalling Equation 1, the detected range is calculated by dividing ΔT by the speed of light after a fixed processing delay of $128\,\mu s$. This delay allows attackers to appear closer than their actual distance by replying to interrogations before a legitimate TCAS system would. Similarly, artificially delaying responses will increase the measured range.

$$T = 128 \cdot 10^{-6} + \frac{c}{2} \cdot (D - D_A) - T_p \tag{5}$$

Equation 5 shows which response time T the attackers should use to make the spoofed aircraft appear at a desired distance D, considering a processing time of T_p.

This equation approximates D_A as a constant. Such simplification is acceptable, as an aircraft approaching at the maximum closure rate accepted in the standard of $1200\,kn$ would move by just $0.08\,m$ during the $128\,\mu s$ round-trip time.

Finally, the assumptions from Section 4.1 state that attackers receive and transmit from a fixed position. Consequently, they cannot alter the injected aircraft bearing. It is worth noting that such a limitation does not influence the attack success rate for TCAS II systems, which do not consider horizontal movements for their corrective maneuvers.

In summary, as long as Equation 5 is non-negative, attackers can control the altitude, speed, and range of the spoofed aircraft, but not its bearing.

Attack effects. In this example, a spoofed intruder is tracked by the victim. The attack assumes SL=2 and uses a fixed range of 0.25 nm. For the altitude, the attack performs an altitude acquisition and simply replays the victim's AC as their own.

Figure 3 shows the effects of the resulting TA on the victim's display. This notification, accompanied by an aural "*TRAFFIC, TRAFFIC*" annunciation, is displayed as a popup over the instrument screen. It contains a flashing yellow banner and a traffic Plan Position Indicator (PPI). This PPI shows the spoofed aircraft bearing and range (yellow circle), augmented with its relative altitude (`00`, i.e., the intruder is at the same altitude).

Figure 3: Avionics displaying a Traffic Advisory.

5.2 Resolution Advisory Injection

A RA injection is a more dangerous variant of the previous attack. In addition to sending the crew an even more urgent alert, a RA effectively allows attackers to impose commands on the victim as pilots are mandated to follow any RA and do so automatically (up to a limit) [47]. This issue becomes more urgent in modern airliners equipped for autonomous RA execution [3], as they allow attackers to manipulate the flight path directly, bypassing human intervention.

Recalling Section 2.2, RAs can be either negotiated between aircraft or fall back to unilateral decisions. In this section, we consider both cases, even though the simpler non-negotiated alternative already allows the attackers to fulfill their goals.

Non-collaborative encounter. This case requires attackers to escalate a TA into a RA by further reducing the separation w.r.t. the previous attack and meeting the additional condition on *ALIM* (see Section 2.2). As such, this attack requires following the same procedure detailed in Section 5.1 with the restriction that altitude should be chosen within *ALIM* of the victim's altitude.

Furthermore, in some situations, attackers can choose which advisory will be issued by reproducing particular encounter geometries. For instance, whenever the victim is climbing or descending, attackers can induce a "*LEVEL OFF*" advisory. To that end, they place the spoofed aircraft at a fixed altitude above/below the victim, forcing the victim's TCAS to abort the current maneuver in order to avoid crossing in front of the intruder stationed around the climb/descent path.

Climbs and descents can be forced whenever the victim is staying at a fixed altitude. Placing a rogue aircraft above will induce a descent, with the opposite inducing a climb.

Other geometries will still trigger a RA, although their outcome might be undefined in real-world scenarios and less useful to attackers aiming to associate a specific action with

their induced RA.

Collaborative encounter. Collaborative encounters are more complex as they require the attackers to perform all of the steps needed by their non-collaborative counterparts and then to take part in the negotiation in order to influence the victim's chosen RA towards their preferred outcome.

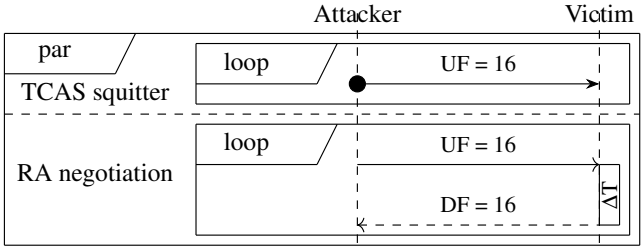

Figure 4: TCAS collaborative encounter tasks.

Figure 4 shows the sequence of additional interactions and messages attackers must exchange to negotiate a RA successfully. These two parallel tasks allow the spoofed aircraft to participate in the RA negotiation. The first operation consists of periodically broadcasting UF=16 TCAS broadcast interrogation squitters, allowing the victim to register the spoofed aircraft as equipped with TCAS. The second operation consists of periodic interrogations with UF=16 TCAS resolution messages with the *Vertical RA Complement* (VRC) field set to the opposite of what maneuver they want to impose, e.g., *don't descend* to order a climb. The victim responds to these interrogations with its currently selected RA sense. Finally, once the RA negotiation is started, attackers need to respond to UF=16 interrogations with *Long air-air surveillance* (DF=16) TCAS coordination replies. Such replies indicate to the victim that the spoofed aircraft's chosen maneuver in the *Active RA* (ARA) field is the opposite w.r.t. the attackers' desired outcome, e.g., an ARA field mentioning *currently descending* will make the victim's TCAS issue a climb.

As mentioned previously, there's a basic arbitration mechanism. By choosing an ICAO address lower than the victim's one, attackers can ensure that their resolution advisory will always prevail over the victim's initial decision.

Attack effects. In this example, attackers are tracked by the victim. Following observation of the victim's squitters, they choose the victim's ICAO address-1 as their spoofed aircraft identity. Attackers assume SL=3 and chose a fixed range of 0.15 nm. Similar to the previous example, they replicate the victim's AC as their own. In addition, they are sending UF=16 coordination messages instructing *don't descend* to the victim's TCAS and indicating a descent as its chosen action in DF=16 responses.

Figure 5 shows the resulting RA on the victim's display. Similar to the previous case, an RA is presented as a popup

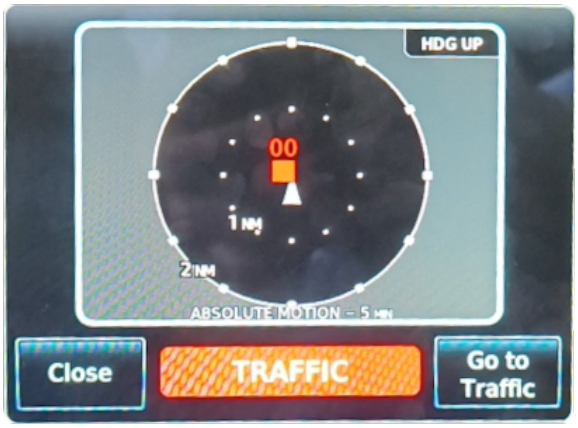

Figure 5: Avionics displaying a Resolution Advisory.

over the instrument screen and is accompanied by an aural indication of its associated action ("*CLIMB CLIMB*"). With respect to the TA case, RAs associate a square symbol to the intruder, and color their indications in red to better convey the urgency of the situation.

5.3 Resolution Advisory Denial-of-Service

According to the protocol, there is an ability to supersede the automatic altitude-based sensitivity level selection with ground-originated Mode S commands.

Specifically, a ground station can issue a *Comm-A Identity Request* (UF=21) directed Sensitivity Level Control (SLC) message containing a candidate sensitivity level. The aircraft, in turn, adjusts its sensitivity settings to the lowest one received from ground stations. This feature was conceived to enable airspace controllers to reduce spurious RAs during procedures with problematic geometries, e.g., parallel approaches. Since SLC messages do not authenticate the issuing ground station, adversaries can manipulate the system into TA-only mode, thereby completely disabling the RA functionality.

(a) Before SLC (b) SLC receipt (c) After SLC

Figure 6: Avionics display during RA DoS.

Figure 6 illustrates the state announcements of the TCAS and the corresponding information displayed on the avionics screen during the attack.

As its initial setup, it is configured to operate normally, i.e., in TA/RA mode (Figure 6a). Attackers identify the victim by listening to its squitters and extracting the corresponding ICAO address. Then, they can impersonate a ground station and send a SLC command directed at the found address. The display shows that the victim has correctly received the com-

mand, as indicated by the *R* in Figure 6b. Once processed by the system, the operating mode is automatically switched to TA-Only mode (Figure 6c), disabling the RA functionality.

If not renewed by a further transmission, standards state that this command should be reset after 240 seconds from its reception to TA/RA mode [37, 2.2.3.10.4]. The above requirement provides adversaries with a significant opportunity, as an attack can be executed and persisted over an extended period with a small number of messages at a very low rate, making detection unlikely.

Moreover, being a *low-and-slow* attack, adversaries can simultaneously engage with multiple aircraft. This capability potentially enables them to further enhance the impact of the attack by easily disabling the RA functionality over a substantial portion of the airspace.

6 TCAS Security Testbed

We design and develop our testbed adhering to the following properties and requirements. A TCAS cybersecurity testbed must be able to facilitate analysis in a *secure* setting, ensuring it does not interfere with the external environment. Additionally, it should be *cost-effective*, eliminating the need to fly an aircraft, and should provide facilities for a consistent and *reproducible* test execution. Finally, its components should be *instrumented* to enable the execution of the experiments and ex-post analysis of the system outputs.

We identify three main components, each composed of multiple constituent elements. The two primary components are the *System Under Test (SUT)* and the entity simulating the TCAS of another aircraft, i.e., the *traffic simulator*. The traffic simulator can also assume the role of the attacker. An additional component, namely the *orchestrator and recorder*, manages the execution of the experiments and records the system outputs. Figure 7 depicts the testbed architecture and its elements. Below, we detail each component.

System Under Test. The core element is a TCAS unit with a radio transponder (XPDR) and processor (PROC). Making this unit operational and testable requires connecting with other elements using avionic data buses. Briefly, the control panel enables its configuration by providing the mode selector switch access. The communication radio, equipped with a headset, and the traffic display provide the outputs from the unit. The former permits the reception of aural warnings, while the latter displays nearby aircraft and related advisories. The execution of the surveillance, TA, and RA functions relies on consistent sensor readings as if the system were in flight. To this aim, an avionic bus simulator provides coherent data, such as the present altitude and velocity. Finally, positioning antennas connected to the radio transceiver inside a Faraday cage prevents unintended transmission of disruptive signals to external sources and complies with the security requirement.

Traffic simulator. This element simulates the presence of other aircraft interacting via radio waves. It contains a workstation (*traffic simulator workstation*) and a linked SDR. On the workstation, an application utilizes the SDR for receiving and transmitting Mode S messages, thereby interacting with the SUT. This application must reliably decode messages, process them, and generate responses coherent with the current condition and desired test scenario. These tasks must be reliably carried out within the $128\,\mu s$ time constraint.

The connected SDR has to feature two independently tunable full-duplex channels, enabling simultaneous downlink and uplink communication. Moreover, it should support highly precise coherent timestamping between reception and transmission and between the two channels. Finally, its sampling rate for each channel has to be higher than its Nyquist rate, i.e. 2 MS/s[2] for the downlink and 4 MS/s for the uplink.

Orchestrator and recorder. The orchestrator and recorder component is responsible for managing the execution of the experiment and capturing measurements from the outputs of the SUT. The *experiment instrumentation workstation* represents the core element.

On the coordination front, it continuously instructs the traffic simulator regarding the specific conditions to simulate. At the same time, it configures the avionic bus simulator to provide data to the avionics buses.

It also captures output data from the TCAS. It records the traffic simulator's current state, representing the desired traffic conditions, and captures the traffic display unit via a camera. Subsequently, image recognition software automatically extracts quantitative data from the raw video feed. This function allows the testbed to align inputs from the traffic simulator with their respective outputs.

7 Implementation

In the following, we will explore the software architecture, adjustments to the operating system, and hardware setup required to overcome the challenges detailed in Section 4.2.

System Under Test. It consists of the elements indicated as "SUT" in Figure 7, with a commercial ARINC 429 bus tester, specifically an Astronics UA2000, as our avionic bus simulator. The TCAS unit consists of a Garmin XPDR model GTX 3000 with GAE 43 and a PROC model GTS 8000. The control panel and traffic display unit, model GTN 750, communication radio, model GMA 342, and antennas, model GA 58, are also from Garmin. Antennas are positioned in a Faraday cage model Ramsey STE3000F2. This setup includes only certified avionic components, all of which, along with their installation, hold a valid airworthiness certificate.

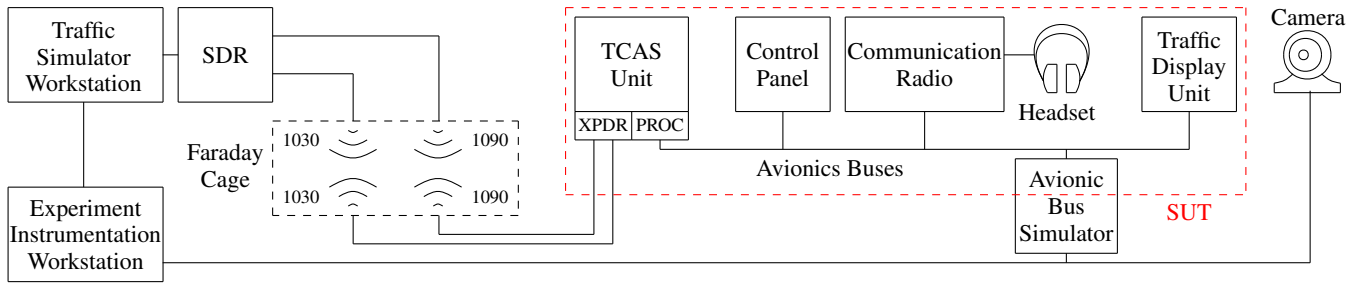

Figure 7: Testbed architecture.

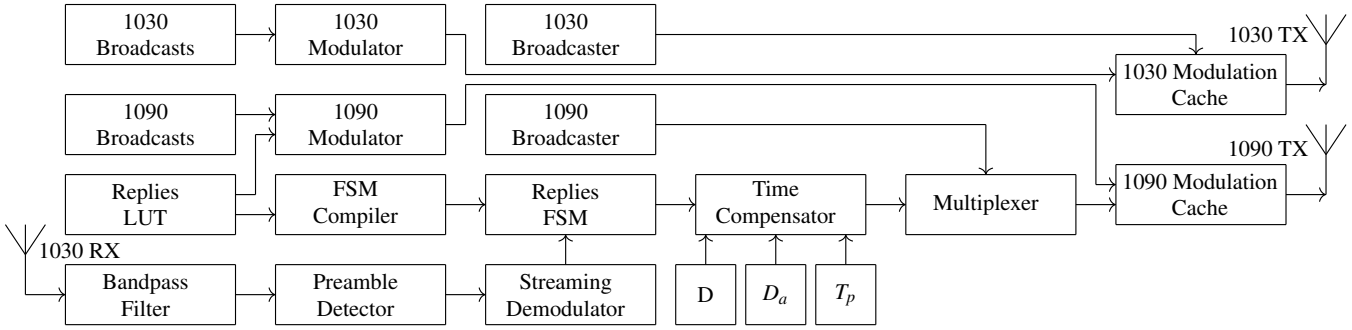

Figure 8: Schematic of the implemented attack software.

Software. A software application was developed to carry out the attacks from this article. Its design aims at handling the interactions described in Section 5, minimizing the reply latency. Its source code consists of 2557 lines of C++ and 15.2K lines of Rust, compiled with GCC 13 and rustc 1.69.

Figure 8 illustrates the architecture of the implemented attack software. Its configuration, continuously updatable from an outside coordinator, is comprised of five parameters:

1. A Replies Lookup Table (LUT) holding UF=x requests and their corresponding DF=x responses, with payload sizes of either 56 or 112 bits.

2. A set of payloads designated for broadcast over the downlink (1090 MHz) frequency (1090 broadcasts), along with their minimum (T_{min}) and maximum (T_{max}) emission intervals.

3. A similar set of payloads for the uplink (1030 MHz) frequency (1030 broadcasts), also governed by T_{min} and T_{max} intervals.

4. The distance D_a between the attacker and the target, as well as the desired distance D of the spoofed aircraft.

5. The estimated processing time T_p.

The system continuously acquires baseband complex quadrature samples at a rate of 20 MS/s from the SDR channel tuned on the uplink frequency. These samples are processed through a Digital Signal Processing (DSP) chain, starting with a bandpass filter and then a fast preamble detector. Given that amplitude modulated pulses identify Mode-S preambles, the detector starts by computing the magnitude of each sample using Single Instruction Multiple Data (SIMD) instructions [25]. It then identifies potential preambles within the received signal by calculating its spectrum using Fast Fourier Transform (FFT) and applying the convolution theorem to calculate its cross-correlation with those characteristic pulses. Candidate preambles are then selected by thresholding their correlation value. The spectrum associated with the pulses is pre-calculated at program startup, halving the number of required FFT calculations.

A streaming demodulator processes each identified preamble and its subsequent samples. This involves frequency and time synchronization, followed by D-BPSK decoding. The resulting bits are then used to navigate the replies Finite State Machine (FSM), looking for prefixes found in the LUT.

Whenever decoded data is unequivocally associated with a LUT entry, the FSM outputs the payload bits to be sent on the downlink frequency. This payload is then timed for transmission based on an interval T relative to the preamble's starting timestamp. T is calculated as given in Equation 5, while T_p is measured by calibrating the system beforehand with self-interrogations.

The system includes two broadcasting elements (1030 and 1090) periodically transmitting payloads (squitters) at random intervals between T_{min} and T_{max}. The broadcasts are managed using a delay queue data structure. A software multiplexer

[2] 1 MS/s = 10^6 samples per second.

handles potential transmission conflicts between the downlink broadcaster and reply transmissions, prioritizing the FSM.

Both the FSM and the broadcasters output bits. These bits are converted into samples using a modulation cache (1030 and 1090 modulation caches) to avoid performing modulation in the attack time-sensitive path. Unique payloads from both the LUT and broadcast lists are pre-modulated by background tasks (1030 and 1090 modulators), which keeps the modulation caches populated and in sync with the configuration. These updates occur without stalling the critical path by leveraging atomic compare and swap operations. Similarly, any update to the LUT prompts a recompilation of the FSM states, which is then introduced into the program's receive path through atomic intrinsics. Finally, the produced samples are sent to the SDR transmission channels.

System configuration and tooling. Our approach involved the strategic implementation of available operating system configurations and the utilization of specialized software tools. We employed a Linux operating system with real-time capabilities (Ubuntu 22.04 with kernel 5.15.0-1032-realtime). Its scheduler was configured to ensure the application was never preempted. We compiled the program with profile-guided optimization and aggressive optimization levels. We pinned the latency-sensitive threads of the application on specific physical cores. Conversely, the kernel threads were moved out of such cores. The application memory was configured to be allocated from huge pages and locked in to prevent swapping. We disabled hardware failure detection interrupts.

Hardware. The attack software ran on an Intel i9-12900k workstation (16 cores running at 5.20GHz). We used an Ettus USRP X300 as our SDR, as it possesses two independently tunable duplex channels with precision timestamping capability, satisfying the requirements outlined in Section 6. Some further tweaks improved the performance: The workstation firmware had all its power-saving features turned off. Unnecessary peripherals and buses in the workstation were removed or disabled to decrease interrupt numbers. The SDR was directly connected to the PCIe bus. We disabled the CPU simultaneous multithreading feature.

8 Evaluation

In this section, we present different experimental evaluations against our TCAS system. We start by assessing the achieved response time and its precision. We also examine whether our implemented optimizations improved these metrics w.r.t. the standalone software. Then, we verify the capability to simulate scenarios with moving aircraft and the accuracy of these simulations. Additionally, we investigated whether stationary injected aircraft scenarios would trigger TAs or RAs and measured the time needed for triggering these alerts. Lastly, we tested the feasibility of the RA DoS attack.

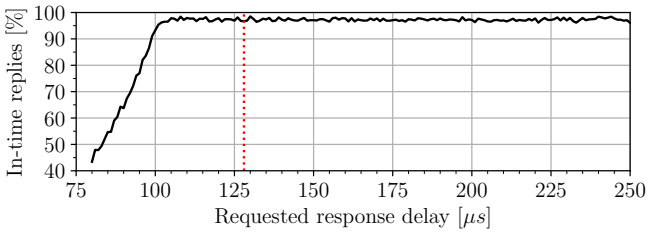

Figure 9: In-time replies as a function of requested delay.

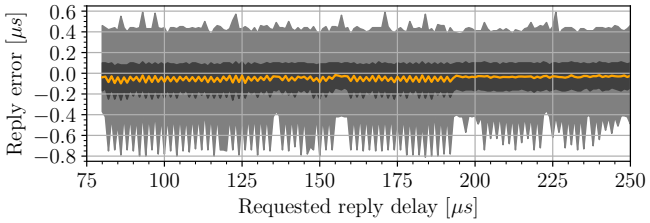

Figure 10: Reply timing precision as requested delay varies.

Reply performance testing. The initial test focused on evaluating timing performance by transmitting a Mode S interrogation, responding to it, and concurrently monitoring the timings using a separate SDR connected to the instrumentation workstation. Response time targets ranging from 80 to 250 μs were tested, with 10,000 samples collected for each target setting. Figure 9 illustrates the percentage of responses received within the desired timeframe. Notably, the success rate begins at approximately 45% around the 80 μs mark and steadily increases to nearly 100% at 100 μs.

Figure 10 illustrates the error envelope, encompassing the min, max, 25th percentile, 75th percentile, and an orange line for the average. These errors consistently stay within a 1.0 μs range, maintaining a stable average around zero. Similarly, Figure 11 displays the Cumulative Distribution Function (CDF) of these errors, demonstrating that the majority remains within a 0.2 μs margin in relation to the desired value.

Figure 12 presents the CDF for jitter, which illustrates the range of delay variations between these responses. The data indicates that the majority of jitter remains below 300 ns.

Effect of optimizations. Figure 13 shows a comparative analysis of response times over 10,000 trials, between the software in its original state and the software after implementing all operating system and hardware modifications. These response times were collected by configuring a target reply time of zero. The analysis reveals that while both distributions exhibit similar patterns, the optimized software's mean response time is reduced by around 45 μs.

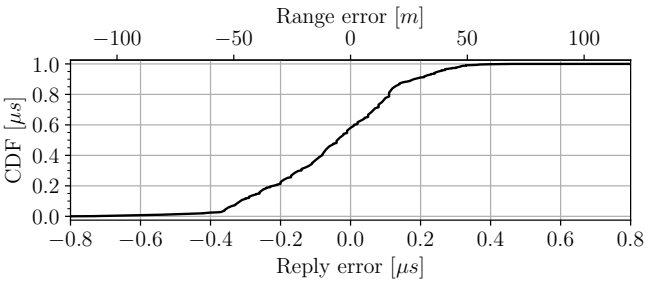

Figure 11: Distribution of reply timing errors.

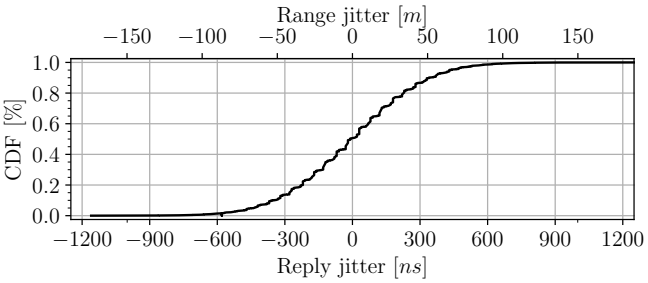

Figure 12: Distribution of reply jitter times.

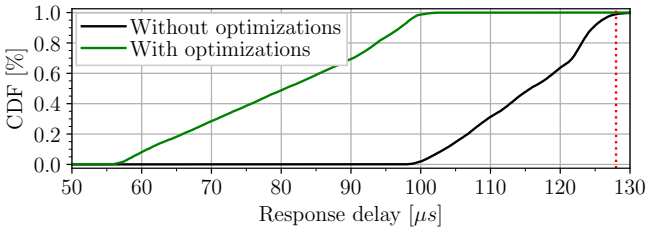

Figure 13: Effects of optimization on reply delay.

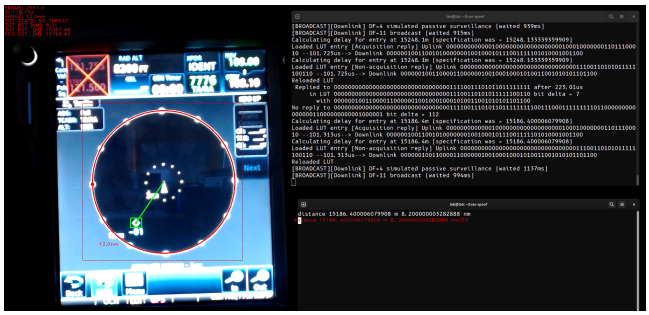

Figure 14: Image recognition applied to the TCAS display.

Moving target scenario. In this test, we simulated an intruder aircraft, commencing the simulation at a distance of 12 nautical miles and advancing towards our location at a steady speed of 100 knots. The data for this experiment was collected by utilizing image recognition techniques on a composite video stream captured from the attack workstation and traffic display. Figure 14 illustrates the computer vision algorithm in action, clearly marking the detected intruder on the avionic display. This procedure was carried out 25 times, resulting in a total of 222,936 data points over a recording duration of 217 minutes. Figure 15 presents a comparison between the desired and observed distance of the moving intruder. The median distance error, depicted in orange, was approx. 125 meters. In Figure 16, we illustrate the distances at which our intruder triggers a TA or an RA. As observed, TAs occur first at approximately 1500 meters, followed by RAs, which occur around the 800-meter mark.

Time delay. For this scenario, we position a static aircraft extremely close, just within the required threshold to initiate a RA. We then monitor the elapsed time of the attack until a TA or RA notification occurs. Figure 17 illustrates the CDF based on 200 repetitions of this scenario. Similar to previous instances, RAs occur after TAs in terms of timing, with TAs taking a maximum of 10 seconds and RAs taking up to 20 seconds to trigger.

RA denial of service. In this test, we transmit the payload detailed in Section 5.3 and interrogate with a separate radio to collect information on the reported sensitivity level. Similar to the display in Figure 18, every sensitivity alteration succeeds immediately upon the TCAS unit receiving the message.

9 Discussion

9.1 Capability Considerations

In relation to the challenges outlined in Section 4.2, our experimental results demonstrate that our implementation consistently and accurately meets the response time requirements using COTS hardware. Furthermore, the applied adjustments and optimizations have effectively and demonstrably decreased this latency. The moving target experiments demonstrated that the precision of reply timing was also maintained over attacks spanning extended periods and distance ranges.

With current hardware and software, attackers can leverage their 28 μs time advantage to fake an aircraft's presence at a distance zero, provided they are within approximately 4.2 km of their victim (Equation 5). With a typical cruise speed of 950 km/h, an aircraft can cover this distance in around 16 seconds, giving attackers a probability of around 80% to successfully execute a RA (Figure 17). As computational power increases, attackers' capabilities will also increase, up to a theoretical limit of 19.186 km, associated with a reply time of zero. Therefore, current cutting-edge COTS hardware has made such attacks feasible, and the attackers' capabilities are expected to continue to grow over time.

As per the RA DoS attack, it can be executed directly and straightforwardly. This simplicity implies that an attacker can initiate this attack even with limited expertise or resources.

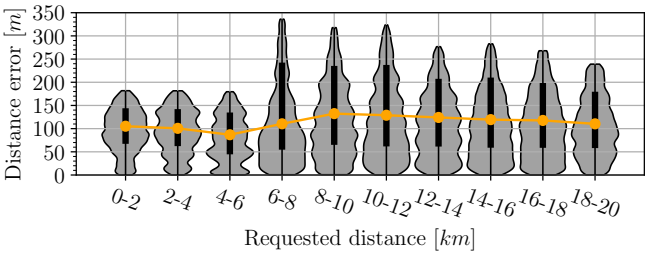

Figure 15: Positioning precision for a moving target.

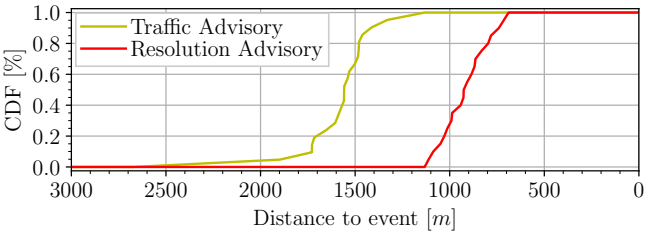

Figure 16: Distance to induce a TA or RA.

9.2 Enabling Factors

In the following, we discuss the enabling factors of the attack based on our experience in its implementation. Briefly, we examine a combination of hardware and software components, and their configurations.

Given that previous attempts used COTS radios attached to workstations similar to ours, we can argue that our success did not stem from any peculiarities in the hardware configuration. In fact, excluding the general requirements set forth in Section 6, our architecture does not rely on any specific radio transceiver.

Instead, the critical difference lies in the software element and our effort to analyze and overcome the limitations of the general-purpose SDR frameworks and standard operating system configurations used in past attempts. Unlike other approaches that utilized multipurpose COTS SDR frameworks like GNURadio [6], which prioritize generality and throughput, our bespoke implementation was designed specifically for low latency and precise timing. For instance, general-purpose frameworks often have buffers built into their processing pipelines, which can improve their throughput, but at the cost of negatively impacting latency — a design choice contrasting with the peculiarities of TCAS communication. Additionally, we made specific changes to the operating system configurations and workstation settings aimed at minimizing delay and jitter. These customizations were crucial in fully exploiting the hardware capabilities to overcome the challenges from Section 4.2, even at the cost of increased energy consumption.

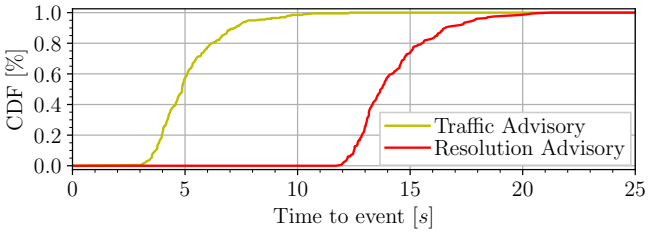

Figure 17: Time to induce a TA or RA.

```
DF0 { df: 0, vertical_status: Airborne, crosslink_capability: Yes, sensitivity_level: L3,
DF0 { df: 0, vertical_status: Airborne, crosslink_capability: Yes, sensitivity_level: L2,
```

Figure 18: Change in reported sensitivity level after RA DoS.

9.3 Limitations

The primary limitations of the attacks detailed in this article are due to their dependency on radio wave transmission. In our controlled experiments, the attacks were conducted without any other device sharing the frequency. However, in actual operational environments, an attacker would face the challenge of developing a more sophisticated receiver system capable of filtering out signals unrelated to the target aircraft.

Transmission-wise, our tests indicated that the attacker's signals must be high enough to be received correctly. This is known as Minimum Trigger Level (MTL) in the standards. In practical terms, the attacker needs a powerful radio transmitter to ensure adequate signal strength and coverage.

Finally, for attacks like the RA injection, it is critical to precisely determine the victim's position and attitude. This task becomes especially challenging due to the high movement speeds associated with aircraft. The complexity escalates if the attacker uses a moving platform, such as a drone, introducing more variables that must be accurately controlled for the attack to succeed.

9.4 Mitigations

Although no practical attacks have been demonstrated in the literature to prove the exploitability of TCAS, general awareness of its possible vulnerabilities has prompted the involved community to evaluate and propose mitigations. These include solutions such as distributed voting algorithms, where aircraft share detected traffic conditions for mutual verification [17], and the integration of encryption to authenticate messages and prevent false aircraft reports [40]. In [17], the authors mention using the radio vertical angle of arrival to verify altitude, providing an additional check before issuing RAs.

Other fields may also inspire viable mitigations. In particular, distance bounding protocols [7, 35] have been used to protect from range spoofing attacks, while Doppler effects have been leveraged to estimate the closure rate of moving transmitters [2, 9, 26, 27, 34].

However, any proposed solution must be evaluated for its

viability. The TCAS specification mandates the acceptance of any correctly addressed message, posing a challenge for the backward-compatible integration of any authorization scheme. As such, enhancements involving protocol changes, while robust, would require widespread adoption and significant changes to existing systems, likely making them infeasible for rapid deployment. Moreover, these changes involve a safety-critical component where false positives are preferred over missed advisories, making them susceptible to downgrade attacks [32], where attackers pose as legacy endpoints unequipped with the security enhancements.

In contrast, modifications that can be implemented per aircraft are progressively deployable and inherently backward compatible. Such solutions involve the integration of physical layer fingerprinting measures on top of the already existing transponder logic. For instance, the proposed vertical angle of arrival calculation is one such method. This enhancement enables cross-checking the declared altitude against the detected angle, quickly identifying discrepancies. Similarly, the analysis of Doppler effects provides an estimated closing rate between aircraft. We deem those two solutions as the most viable as they can significantly increase the difficulty for attackers to spoof aircraft positions effectively. For example, attackers attempting to induce a "DESCEND" RA would be constrained to position their antenna above the victim, inhibiting almost every attack carried out from the ground. Similarly, fooling the Doppler-based solution would require emulating the frequency shift as perceived by the victim, requiring solving the positioning problem also for the derivative of range, i.e., closure rate.

Given our findings, authorities must avoid relying solely on self-reported information in future collision avoidance standards, including ACAS X, until robust security measures are in place. As such, we urge standards bodies to explore incorporating modern wireless security technologies into future collision avoidance systems before they become commonplace. Therefore, an ongoing discussion [51] is contemplating a backward-compatible modification to allow additional data to be transmitted in each Mode-S downlink message, possibly enabling the inclusion of dedicated cryptographic authentication fields.

Interestingly, the ACAS X draft standards inadvertently address the vulnerability to RA DoS attacks by not depending on the SL parameter for their calculations.

Finally, it is worth considering procedural approaches such as awareness for pilots and controllers for the (highly unusual) case of an attack as discussed in [47].

9.5 Ethical Considerations

We disclosed our work and this associated article on submission, and the process is still ongoing. The manufacturers we contacted acknowledged the attack and understood that the vulnerabilities presented in the article are not specific to their devices. They deemed no action necessary on their end, as the attack affected any standards-compliant system. Specifically, Garmin recognized it as a vendor-independent vulnerability. Airbus had prior knowledge of one of the attacks (RA DoS) and had already notified their contacts. The division of Leonardo that manufactures TCAS equipment organized a dedicated meeting to discuss the details of the attack. We are working on a formal collaboration involving our institutions to further test their systems with our tools. Several European national and supra-national authorities and pilot organizations were informed of the results and are in contact with the authors. A dedicated meeting was held with EASA, and they will incorporate our feedback as part of their future decisions on this topic. The Aviation ISAC has forwarded our paper to relevant stakeholders in their disclosure process.

Additionally, we have disclosed our findings to the FAA, Boeing, Pilatus Aircraft, and Thales, although we have yet to receive an acknowledgment from them. We have asked the United States CISA CVD to facilitate further communication.

TCAS will be in use for a long time and no changes are expected. We hope that our work will lead to improvements for ACAS X and future collision avoidance standards.

Attacks on TCAS have been hypothesized before, but by proving their existence and defining the practical parameters and requirements, we hope to improve awareness in aviation. Military actors are likely to have access to toolsets able to attack TCAS, so we need to think about processes and procedures now to not be surprised when an attack happens.

We created our experimental setup such that we followed all local laws regarding radio frequency communication, taking care that no signals leaked outside the setup and the building.

We have chosen not to include the source code for executing the attack in our artifact release, as the potential negative impacts of enabling such capabilities outweigh its value to the research community.

10 Conclusions

This paper presents three wireless attacks against the TCAS. The first two attacks aim at disrupting pilots' situational awareness by inducing a traffic alert. Our study presents the first successful implementation overcoming the well-documented challenges in the realm of TCAS range estimation, which has long served as a de facto physical-layer security feature. Our experimental evaluations demonstrate that we can surpass the required performance metrics, leaving a margin of $28\,\mu s$. We can arbitrarily position the spoofed aircraft in both dynamic and stationary scenarios, consistently inducing the intended effects. The third novel attack causes a DoS simply by using features found in the TCAS protocol. We also discussed potential mitigations but the practical implementation of a countermeasure is still an open problem. We hope that demonstrating the feasibility of the attack will enhance aviation awareness and drive advancements for ACAS X.

Availability

The primary data sets used in this study, including all raw measurements, as well as the scripts used for analysis, are openly available for review and further research. Additionally, supplementary videos providing visual evidence and additional context are also provided.

All these materials can be accessed at [29].

We encourage readers to explore these resources for a more comprehensive understanding of our research and findings.

Acknowledgments

We are grateful for the support of Q.C.M. quality control management AG, who assisted in assembling the testbed. This work was partially funded by the NextGenerationEU project "Security and Rights in CyberSpace" (SERICS). It was carried out while Giacomo Longo was enrolled in the Italian National Doctorate on Artificial Intelligence run by the Sapienza University of Rome in collaboration with the University of Genoa. Giacomo would like to extend his sincere thanks to Daniel, Giorgio, Ivo, Michiel, and his co-authors for their emotional support throughout the project execution.

References

[1] COMMISSION REGULATION (EU) No 1332/2011 - Laying down common airspace usage requirements and operating procedures for airborne collision avoidance. *Official Journal of the European Union*, 56(L 336), December 2011.

[2] Chris Ashton, Alan Shuster Bruce, Gary Colledge, and Mark Dickinson. The search for mh370. *The Journal of Navigation*, 68(1):1–22, 2015.

[3] Christophe Baroux, Florence Cardona, Cédric Corral, Xavier Durepaire, Maria Luisa Lopez Villarejo, and Christine Villeneuve. Safe Handling of TCAS Alerts. Technical report, Airbus, 2021. https://safetyfirst.airbus.com/safe-handling-of-tcas-alerts/, last accessed on 02/01/24.

[4] Paul M Berges, Basavesh Ammanaghatta Shivakumar, Timothy Graziano, Ryan Gerdes, and Z Berkay Celik. On the feasibility of exploiting traffic collision avoidance system vulnerabilities. In *2020 IEEE Conference on Communications and Network Security (CNS)*, pages 1–6. IEEE, 2020.

[5] Paul Martin Berges. *Exploring the vulnerabilities of traffic collision avoidance systems (TCAS) through software defined radio (SDR) exploitation*. PhD thesis, Virginia Tech, 2019.

[6] Eric Blossom. Gnu radio: tools for exploring the radio frequency spectrum. *Linux journal*, 2004(122):4, 2004.

[7] Stefan Brands and David Chaum. Distance-bounding protocols. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 344–359. Springer, 1993.

[8] CENIPA. Final Report. Technical Report A-00X/CENIPA/2008. https://skybrary.aero/sites/default/files/bookshelf/546.pdf, last accessed on 29/01/24.

[9] Lin Cheng, Benjamin Henty, Fan Bai, and Daniel D. Stancil. Doppler spread and coherence time of rural and highway vehicle-to-vehicle channels at 5.9 ghz. In *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pages 1–6, 2008.

[10] Andrei Costin and Aurélien Francillon. Ghost in the air (traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices. *black hat USA*, 1:1–12, 2012.

[11] EUROCAE. Minimum Operational Performance Standards for Secondary Surveillance Radar Mode S Transponders, May 2011. ED-73E.

[12] EUROCONTROL. Ta-only mode. *ACAS Bulletin*, (26), May 2022.

[13] Federal Aviation Administration. Collision Avoidance Systems. *Federal Register*, 68(62):15884–15904, April 2003.

[14] German Federal Bureau of Aircraft Accidents Investigation. Investigation Report. Technical Report AX001-1-2/02, May 2004. https://skybrary.aero/sites/default/files/bookshelf/414.pdf, last accessed on 29/01/24.

[15] Timothy Michael Graziano. *Establishment of a Cyber-Physical Systems (CPS) Test Bed to Explore Traffic Collision Avoidance System (TCAS) Vulnerabilities to Cyber Attacks*. PhD thesis, Virginia Tech, 2021.

[16] Edan Habler, Ron Bitton, and Asaf Shabtai. Assessing Aircraft Security: A Comprehensive Survey and Methodology for Evaluation. *ACM Computing Surveys*, 56(4):1–40, November 2023.

[17] John Hannah, Robert Mills, Richard Dill, and Douglas Hodson. Traffic collision avoidance system: false injection viability. *The journal of supercomputing*, pages 1–24, 2021.

[18] John W Hannah. A cyber threat taxonomy and a viability analysis for false injections in the TCAS. 2021.

[19] ICAO. *Doc 9863 - Airborne Collision Avoidance System (ACAS) Manual*. 1 edition, 2006.

[20] ICAO. Annex 10 to the convention on International Civil Aviation - Aeronautical Telecommunications. 3 - Communication Systems, 2007.

[21] ICAO. *Doc 9871-AN/464 - Technical Provisions for Mode S Services and Extended Squitter*. 2 edition, 2012.

[22] ICAO. Annex 10 to the convention on International Civil Aviation - Aeronautical Telecommunications. 4 - Surveillance and Collision Avoidance, 2014.

[23] ICAO. Annex 6 to the convention on International Civil Aviation - Operation of Aircraft. 1 - International Commercial Air Transport - Aeroplanes, 2022.

[24] ICAO. Annex 8 to the convention on International Civil Aviation - Airworthiness of Aircraft. 2022.

[25] Intel. Intrinsics guide. https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html, 2023.

[26] K. Jakus and D.S. Coe. Speed measurement through analysis of the doppler effect in vehicular noise. *IEEE Transactions on Vehicular Technology*, 24(3):33–38, 1975.

[27] Branislav Kusy, Akos Ledeczi, and Xenofon Koutsoukos. Tracking mobile nodes using rf doppler shifts. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, page 29–42, New York, NY, USA, 2007. Association for Computing Machinery.

[28] Alex Lomas. DEFCON 28 Aerospace Village: ILS and TCAS Spoofing. Technical report, 2020. https://www.pentestpartners.com/security-blog/ils-and-tcas-spoofing/, last accessed on 02/01/24.

[29] Giacomo Longo, Strohmeier Martin, Enrico Russo, Alessio Merlo, and Vincent Lenders. Dataset for "On a Collision Course: Unveiling Wireless Attacks to the Aircraft Traffic Collision Avoidance System (TCAS)". May 2024. https://doi.org/10.5281/zenodo.11351913.

[30] Georgia Lykou, George Iakovakis, and Dimitris Gritzalis. *Aviation Cybersecurity and Cyber-Resilience: Assessing Risk in Air Traffic Management*, page 245–260. Springer International Publishing, 2019.

[31] Nils Mäurer, Tobias Guggemos, Thomas Ewert, Thomas Gräupl, Corinna Schmitt, and Sophia Grundner-Culemann. Security in Digital Aeronautical Communications A Comprehensive Gap Analysis. *International Journal of Critical Infrastructure Protection*, 38:100549, September 2022.

[32] MITRE. T1562.010 - impair defenses: Downgrade attack.

[33] National Transportation Safety Board. Aviation Investigation Final Report. Technical Report OPS11IA410. https://skybrary.aero/sites/default/files/bookshelf/1967.pdf, last accessed on 29/01/24.

[34] Alxender Paier, Johan Karedal, Nicolai Czink, Helmut Hofstetter, Charlotte Dumard, Thomas Zemen, Fredrik Tufvesson, Andreas F. Molisch, and Christoph F. Mecklenbrauker. Car-to-car radio channel measurements at 5 ghz: Pathloss, power-delay profile, and delay-doppler spectrum. In *2007 4th International Symposium on Wireless Communication Systems*, pages 224–228, 2007.

[35] Kasper Bonne Rasmussen and Srdjan Capkun. Realization of RF distance bounding. In *19th USENIX Security Symposium (USENIX Security 2010)*, 2010.

[36] RTCA. Traffic Alert and Collision Avoidance System (TCAS) I Functional Guidelines, May 1983.

[37] RTCA. DO-185B - Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II), June 2008. Version 7.1.

[38] RTCA. DO-185B - Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II), June 2008. Version 7.1, Volume II.

[39] RTCA. DO-181E - Minimum Operational Performance Standards for Air Traffic Control Radar Beacon System / Mode Select (ATCRBS / Mode S) Airborne Equipment. Technical report, March 2011.

[40] Mayara Lopes Salgado and Marcelo Santiago de Sousa. Cybersecurity in Aviation: the STPA-Sec Method Applied to the TCAS Security. In *2021 10th Latin-American Symposium on Dependable Computing (LADC)*. IEEE, November 2021.

[41] Harshad Sathaye, Guevara Noubir, and Aanjhan Ranganathan. On the Implications of Spoofing and Jamming Aviation Datalink Applications. In *Proceedings of the 38th Annual Computer Security Applications Conference*. ACM, December 2022.

[42] Harshad Sathaye, Domien Schepers, Aanjhan Ranganathan, and Guevara Noubir. Wireless attacks on aircraft landing systems. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 295–297, 2019.

[43] Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. Experimental Analysis of Attacks on Next Generation Air Traffic Communication. In *Applied Cryptography and Network Security*, pages 253–271. Springer Berlin Heidelberg, 2013.

[44] Matthias Schäfer, Xavier Olive, Martin Strohmeier, Matthew Smith, Ivan Martinovic, and Vincent Lenders. OpenSky Report 2019: Analysing TCAS in the real world using big data. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pages 1–9. IEEE, 2019.

[45] Joshua Smailes, Daniel Moser, Matthew Smith, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. You Talkin' to Me? Exploring Practical Attacks on Controller Pilot Data Link Communications. In *Proceedings of the 7th ACM on Cyber-Physical System Security Workshop*, CPSS '21, page 53–64, New York, NY, USA, 2021. Association for Computing Machinery.

[46] Matthew Smith, Daniel Moser, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. Undermining Privacy in the Aircraft Communications Addressing and Reporting System (ACARS). *Proceedings on Privacy Enhancing Technologies*, 2018(3):105–122, April 2018.

[47] Matthew Smith, Martin Strohmeier, Jonathan Harman, Vincent Lenders, and Ivan Martinovic. A View from the Cockpit: Exploring Pilot Reactions to Attacks on Avionic Systems. In *Proceedings 2020 Network and Distributed System Security Symposium*. Internet Society, 2020.

[48] Matthew Smith, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. Understanding realistic attacks on airborne collision avoidance systems. *Journal of Transportation Security*, 15(1-2):87–118, February 2022.

[49] Martin Strohmeier, Matthias Schafer, Rui Pinheiro, Vincent Lenders, and Ivan Martinovic. On Perception and Reality in Wireless Air Traffic Communication Security. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2016.

[50] Swiss Accident Investigation Board. Final Report. Technical Report 2211, April 2013. https://www.sust.admin.ch/inhalte/AV-berichte/2211_e.pdf, last accessed on 29/01/24.

[51] SESAR 3 Joint Undertaking. Sesar 2020 pj.14-w2-84d - phase overlay for ads-b - contextual note. Technical report, 2022. https://www.sesarju.eu/sesar-solutions/phase-overlay-ads-b, last accessed on 20/5/2024.